PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F — A2

(11) International Publication Number: WO 98/37475

(43) International Publication Date: 27 August 1998 (27.08.98)

(21) International Application Number: PCT/US98/02334

(22) International Filing Date: 6 February 1998 (06.02.98)

(30) Priority Data:
60/039,320 7 February 1997 (07.02.97) US

(71) Applicant: MORPHOLOGIC, INC. [US/US]; 131 Daniel Webster Highway #470, Nashua, NH 03060 (US).

(72) Inventors: MARTUSCELLO, Anthony, R.; 11 Oak Street, Dover, NH 03820 (US). BARBA, Alexandru; 131 Daniel Webster Highway #405, Nashua, NH 03060 (US). BOX, Brian; 131 Daniel Webster Highway #327, Nashua, NH 03060 (US). FURCINITI, Charles; 25 Proctor Road, Bedford, NH 03110 (US).

(74) Agents: BOURQUE, Daniel, J. et al.; Law Offices of Daniel J. Bourque, PA, Suite 303, 835 Hanover Street, Manchester, NH 03104 (US).

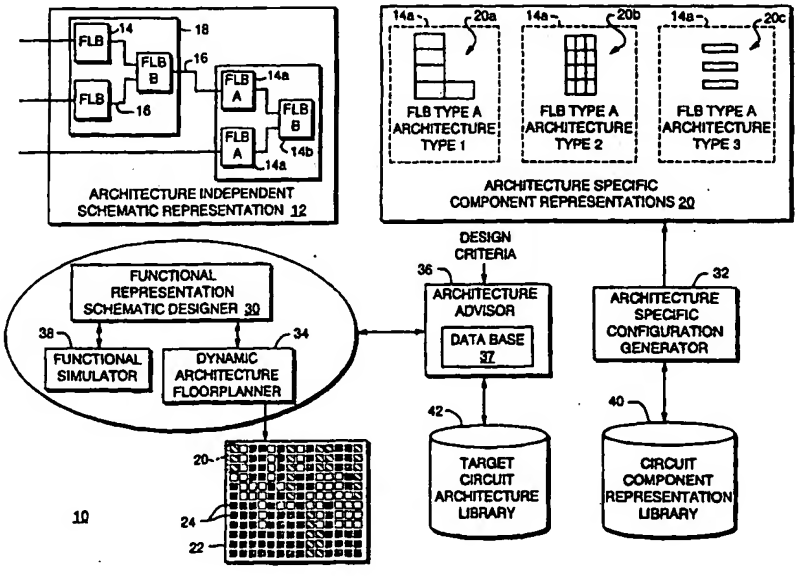
(81) Designated States: CA, JP, Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: SYSTEM AND METHOD FOR DESIGNING ELECTRONIC CIRCUITS

(57) Abstract

A computer–implemented electronic design system (10) is used to design electronic circuits, such as field programmable gate arrays (FPGAs) and other complex logic circuits. The system allows a user to create an architecture independent schematic representation (12) of a circuit by selecting, arranging and interconnecting functional representations (14) of the circuit components. The system automatically configures the functional representations (14) for a selected target circuit architecture by generating architecture specific representations (20) of the circuit components for placement on the selected target architecture representation (22). The architecture specific representations (20) placed on the selected target architecture representation (22) can be moved to new locations or to another selected target architecture representation (22) and are dynamically re–configured for the new location or new target architecture. The system creates a database (37) of design parameters and monitors the design process, providing real time design rules checking during creation of the schematic representation and placement of the architecture specific representations on the target architecture representation. If any design errors are detected, the system warns the user and suggests a design change and/or automatically corrects the error, e.g., by dynamically re–configuring the architecture specific representations (20). The system will also simulate the logic functionality of the schematic representation in a multi–clock, half clock functional simulator environment.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

SYSTEM AND METHOD FOR DESIGNING ELECTRONIC CIRCUITS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application Serial No. 60/039,320 filed February 7, 1997, fully incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to systems and methods for designing electronic circuits and in particular, to a computer-implemented electronic design system and method in which an architecture independent schematic representation of electronic circuit components is automatically configured for placement on a selected target circuit architecture.

BACKGROUND OF THE INVENTION

Electronic circuits, such as field programmable gate arrays (FPGAs) and other types of programmable logic devices, have become increasingly complex with advances in technology. An FPGA, for example, typically includes an array of thousands of logic elements, such as gates, look-up tables, flip-flops and the like, on a single integrated circuit (IC) chip. The logic circuits can be programmed by an end user at their site according to the desired operation to be performed. The ability to program FPGAs to perform complex logic operations at high speeds with a very small, high-density electronic circuit has made these devices popular for a number of different applications. FPGAs, for example, are often used in various logic applications, such as digital signal processing, custom computing, and logic emulation.

Numerous different types of FPGA architectures are available from different vendors, such as Xilinx®, Lucent Technologies, and National Semiconductor Clay Logic. In FPGA architectures, the logic elements are physically grouped into sites, which are physical locations on the chip that include a group of logic elements. FPGA architectures can

be broken down into an array of sites that represent the lowest logical granularity or breakdown within the FPGA architecture. One type of FPGA architecture available from Xilinx®, for example, are broken down into Configurable Logic Blocks (CLBs) and Input Output Blocks (IOBs). The grouping of programmable logic elements and the granularity of the FPGA architectures can vary greatly between integrated circuit different vendors.

Because of the large number of logic elements on these high-density logic circuits, electronic design automation (EDA) tools have been developed to facilitate programming of FPGAs and other similar electronic devices. Using EDA software the user can design the algorithm to be performed by the circuit on a higher level, and then translate the algorithm to a lower level, such as the register transfer level (RTL). EDA software can also perform a simulation to verify the design prior to physically implementing the design on the actual circuit architecture of the FPGA.

The conventional EDA systems and methods, however, have a number of limitations. As the complexity of the logic circuits increase, so does the difficulty of ensuring that the design is correct, for example, that the circuit components are properly arranged and connected in the circuit architecture. Not only is the complexity of the logic circuits drastically increasing, but also there is an increasing demand for quick design cycles. The difference in granularity between the different architectures available from different vendors has also challenged the conventional circuit design techniques.

One method of designing FPGAs and other similar electronic circuits is known as schematic entry. A schematic design of the circuit is created for a particular type of target circuit architecture produced by a specific vendor. When the schematic design for the circuit is complete, simulation verification is performed. If any design errors are detected, the schematic design must be re-designed. This iterative process continues until no design

errors are detected during the simulation verification. The schematic design is then placed and routed within the pre-selected circuit architecture, i.e., the physical locations and interconnections on the pre-selected circuit architecture are determined. A timing verification is then performed, and if timing errors are detected, the placement and routing must be modified or the schematic design of the circuit must be redesigned.

One problem with this type of design system and method is that the schematic design entry is created based upon a specific target circuit architecture. If the same or similar algorithm needs to be re-targeted to a different circuit architecture, a new schematic design must be created based upon the speed and timing requirements for that type of target circuit architecture. Thus, the user is locked into one specific circuit architecture from the beginning of the design process.

Another type of system and method for designing FPGAs and other similar logic devices is commonly known as behavioral synthesis. This technique involves creating a high level description of the algorithm using a hardware description language (HDL), such as the VHDL or Verilog. The synthesis of this high level description of how the circuit behaves creates a list of the logic gates and the interconnections to be implemented on the circuit architecture. Simulation verification is then performed to detect any design errors. If design errors are detected, the designer must revise the HDL description accordingly. The synthesized design is then placed and routed on a selected target circuit architecture. A timing verification is performed to determine if any timing errors exist in the electronic circuit implemented on the selected target circuit architecture. If a timing error is detected, the circuit design must either be re-routed or the circuit must be redesigned using the HDL.

While behavioral synthesis appears to provide a more flexible architecture independent approach, existing behavioral synthesis tools have had limited success in FPGA

design due to the vast differences in granularity between the different architectures provided by different vendors. Thus, neither of these existing electronic circuit design tools allow a user to easily transfer electronic circuit designs from one type of circuit architecture to another. The existing design tools also do not allow mapping across different architectures in a heterogeneous floorplanning environment. Another problem that exists with both of these electronic circuit design systems and methods is that different software design tools are required for the initial high-level design entry and the place and route functions. The different types of software tools that are required for the complete design process often have different database formats and do not efficiently interact during the design process.

Also, these existing FPGA design techniques do not perform a simulation verification until after the high level description is completed, e.g. after the HDL description is entered or after the schematic design is entered. Thus, the designer must spend significant time and effort in preparing the initial design without any feedback as to whether or not design errors might exist. Any problems that are found during the verification typically require a redesign. Also, routing and timing verification does not occur until even later in the design cycle after placement and routing is completed. Any routing or timing errors could also result in having to redesign from the beginning of the design cycle. The user then will not know whether redesign has corrected the routing/timing errors until the placement and routing of the modified design is complete. Thus, these design systems and methods often require multiple lengthy iterations before a complete, optimized circuit design is ready to be physically implemented.

Accordingly, a need exists for a system and method for designing electronic circuits that is capable of automatically configuring a high level schematic design to a number of different types of target circuit architectures having different granularities. A need also exists for an

integrated electronic design system and method in which the different tools of the system work together to efficiently design the electronic circuit while performing design verification or checking in real time throughout the design cycle. A need further exists for an electronic design system that verifies and/or optimizes placement and routing of the circuit components in real time as the circuit components are moved to different locations in the target circuit architecture or moved from one target architecture to a different target architecture.

SUMMARY OF THE INVENTION

The present invention features a computer-implemented electronic design system for designing an electronic circuit having at least one circuit component within a target circuit architecture. The system comprises an architecture advisor, for creating a database of design criteria, and for monitoring and managing design of the electronic circuit using the design criteria, and a functional representation schematic designer, for creating and displaying an architecture independent schematic representation of the electronic circuit including a functional representation of each circuit component in the electronic circuit. The system further includes an architecture specific configuration generator, for generating at least one architecture specific representation corresponding to the functional representation of each electronic circuit component when implemented on at least one target circuit architecture, and a dynamic architecture floorplanner, for creating and displaying at least one architecture specific schematic representation of the electronic circuit The architecture specific schematic representation includes a target circuit architecture representation having an array of architecture sites. The architecture specific configuration generator generates an architecture specific representation configured for the architecture sites of the target circuit architecture representation when the functional

representation of each circuit component is placed at a location on the target circuit architecture representation. The architecture advisor is coupled to and communicates with the functional representation schematic designer, the architecture specific configuration generator, and the dynamic architecture floorplanner.

The preferred embodiment of the system includes a functional simulator responsive to the architecture advisor, for simulating functionality of the architecture independent schematic representation of the electronic circuit. The functional simulator is capable simulating functionality using multiple clock speeds and half clock cycles, as well as performing dynamic pipeline realignment.

The system also preferably includes a circuit component representation library including a plurality of predefined functional representations of circuit components, and including a plurality of architecture specific representations associated with each of the functional representations. The functional representation schematic designer allows the user to select and arrange selected ones of the predefined functional representations and to provide interconnections between each of the selected predefined functional representations. In one example, the functional representation schematic designer is used to create a hierarchy of functional representations. The architecture specific configuration generator preferably creates a cache for storing a plurality of architecture specific representations corresponding to the selected functional representations in the architecture independent schematic representation.

The preferred embodiment architecture advisor that includes a design rules checker, for performing design rules checking in real time as the electronic circuit is designed, and wherein the architecture advisor informs the user if a design error is detected. The preferred embodiment of the architecture advisor provides automatic correction of the design in response to a design error, for example, by causing the architecture specific

representations to be re-configured.

The preferred embodiment of the dynamic architecture floorplanner notifies the architecture advisor of a location at which a functional representation is placed on the selected target architecture such that the architecture advisor causes the architecture specific configuration generator to generate an architecture specific representation appropriate for the location on the target circuit architecture representation. The design rules checker of the architecture advisor preferably checks the location of the architecture specific representation, and if a design error is detected, the architecture advisor causes the architecture specific representation to re-configure to correct the error.

The dynamic architecture floorplanner also preferably allows the user to move the architecture specific representations on the target architecture representation to a new location on the target architecture representation or to another selected target architecture representation. The architecture advisor, in response to movement of the architecture specific representation, causes the architecture specific representation to dynamically re-configure. The dynamic architecture floorplanner preferably displays the changes to the architecture specific representations as they are reconfigured.

The present invention also features a computer-implemented system for designing an architecture independent schematic representation of a logic circuit comprising the architecture advisor, a circuit component representation library, and the functional representation schematic designer. The present invention also features a computer-implemented system for creating an architecture specific schematic representation of an electronic circuit comprising the architecture advisor, the architecture specific representation generator, and the dynamic architecture floorplanner.

The present invention also features a computer-readable medium for providing at least one electronic

circuit component representation for use in a computer-implemented electronic design system. The computer-readable medium comprises architecture independent functional data stored thereon, for defining at least one function to be performed by at least one electronic circuit component independent of the target circuit architecture. The computer-readable medium further comprises architecture specific configuration data stored thereon with the architecture independent functional data for each electronic circuit component, for defining a plurality of implementations of the function on a plurality of target circuit architectures. Also stored on the computer-readable-medium is at least one configuration routine that responds to a request from the computer-implemented electronic design system indicating a selected location for the electronic circuit component on the target architecture, for configuring the architecture specific configuration data to create an architecture specific representation appropriate for the selected location of the electronic circuit component on the target architecture.

The present invention also features a method of designing electronic circuits independent of the system defined above. The method comprises the steps of: capturing design parameters; selecting and displaying a functional representation of each electronic circuit component, wherein the functional representation represents an architecture independent function performed by each electronic circuit component; creating an architecture independent schematic representation of the electronic circuit having the electronic circuit design parameter by arranging and interconnecting the functional representation of each electronic circuit component; selecting and displaying a target architecture representation for the electronic circuit, the target architecture representation including an array of architecture sites; creating an architecture specific schematic representation from the functional representations of each electronic circuit component, wherein the architecture specific schematic

representation includes an architecture specific representation of each circuit component implemented on the selected target architecture representation, and wherein the architecture specific representation is configured for the architecture sites on the selected target architecture; and checking design rules in real time during designing the electronic circuit according to the design parameters.

According to one method the step of selecting the functional representations includes selecting at least one large scale functional representation of a combination of circuit components. The method can also include creating and storing large scale functional representations of a combination of circuit components.

According to the preferred method, the checking of design rules includes monitoring the selection, arrangement and interconnection of the functional representation, and checking for inconsistencies with the design parameters. In one example, the method further includes the step of simulating functionality of the architecture independent schematic representation.

According to the preferred method, the step of creating an architecture specific schematic representation from the functional representations includes moving each functional representation to a location on the selected target architecture representation. The architecture specific representation is dynamically configured based upon the architecture sites at the location on the target architecture representation. The step of checking design rules in real time preferably includes checking the location of the architecture specific representation against the design parameters, and dynamically re-configuring the architecture specific representation in response to detecting an error. The architecture specific representation can be dynamically re- configured based upon a proximity of the architecture specific representation to another architecture specific representation on the target architecture representation or base upon a set up time error detected when the architecture specific

representation is placed at the location.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features and advantages of the present invention will be better understood by reading the following detailed description, taken together with the drawings wherein:

Fig. 1 is a schematic block diagram of a computer implemented electronic design system, according to the present invention;

Fig. 2 is a schematic block diagram of the computer-implemented electronic design system, according a preferred embodiment of the present invention;

Fig. 3 is a screen image of a user interface created by the architecture advisor for capturing design parameters, according to one embodiment of the computer implemented electronic design system of the present invention;

Fig. 4 is a screen image of a user interface created by the functional representation schematic designer for creating the architecture independent schematic representation, according to one embodiment of the computer implemented electronic design system of the present invention;

Fig. 5 is a screen image of a user interface created by the dynamic architecture floorplanner for creating the architecture specific schematic representation, according to one embodiment of the computer implemented electronic design system of the present invention;

Fig. 6 is a screen image of the functional representation schematic designer user interface together with the dynamic architecture floorplanner user interface, according to one embodiment of the computer implemented electronic design system of the present invention;

Figs. 7A-7C are screen images of the dynamic re-configuration of architecture specific representations as they are moved on the target architecture representation, according to one embodiment of the computer implemented electronic design system of the present invention;

Figs. 8A and 8B are screen images of a design error notification provided to the user in response to design rules checking, according to one embodiment of the computer implemented electronic design system of the present invention;

Figs. 9A and 9B are screen images of the dynamic re-configuration of an architecture specific representation in response to an error detected by design rules checking, according to one embodiment of the computer implemented electronic design system of the present invention; and

Figs. 10A and 10B are screen images of the dynamic re-configuration of a group of architecture specific representations, according to one embodiment of the computer implemented electronic design system of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The computer-implemented electronic design system 10, Fig. 1, according to the present invention, is used to design complex electronic circuits using a hierarchy of architecture independent schematic representations that are mapped to one or more selected target circuit architectures. Although the exemplary embodiment focuses on FPGA design, the present invention contemplates using this system and method to design other types of complex electronic circuits, such as application specific integrated circuits (ASICs). In the exemplary embodiment, the computer-implemented electronic design system 10 is implemented as software on a personal computer (PC), although the present invention contemplates other implementations including software, hardware, firmware, or any combination thereof.

According to the broad concept of the present invention, the computer-implemented electronic design system 10 is used to create an architecture independent schematic representation 12 of an electronic circuit to be designed according to a specified algorithm and design criteria. The architecture independent schematic representation 12 includes functional representations or functional logic

blocks (FLBs) 14 that represent the function of the circuit components or logic elements (e.g., adders, multipliers, multiplexers, registers, and the like) in the electronic circuit. The functional representations 14 are arranged and interconnected with net segments 16 according to the algorithm to be performed by the circuit. The architecture independent schematic representation 12 can preferably include a hierarchy of functional representations 14 organized into one or more groups 18.

Each of the functional representations 14 has corresponding architecture specific representations 20 that represent the implementation or mapping of the functional representations 14 on different target circuit architectures. A functional representation 14a of an adder, for example, will have different configurations 20a-20c when mapped to a Xilinx® 4K Series architecture, a Xilinx® 6200 series architecture, and a Lucent® 2C Series architecture. The appropriate architecture specific representations 20 corresponding to the functional representations 14 in the architecture independent schematic representation 12 are placed on a selected target circuit architecture representation 22 including an array of architecture sites 24 to create an architecture specific schematic representation. Each architecture specific representation 20 is configured to occupy at least a portion of the architecture sites 24 and is located such that the design criteria for the electronic circuit are met.

The computer-implemented electronic design system 10 includes a number of components or software tools. A functional representation schematic designer 30 creates and displays the architecture independent schematic representation 12 by allowing the user to select, arrange, and interconnect the functional representations or FLBs 14.

An architecture specific configuration generator 32 generates and manages the architecture specific representations 20 corresponding to the FLBs 14 selected and arranged in the architecture independent schematic

representation 12. A dynamic architecture floorplanner 34 creates and displays the architecture specific schematic representation including the selected target architecture representation 22 with the appropriate architecture specific representations 20 placed and configured onto the architecture sites 24 at the appropriate locations.

An architecture advisor 36 receives design criteria, for example, input by the user, and creates a database 37 of design criteria and monitors and manages design of the circuit according to the design criteria. The architecture advisor 36 preferably interacts with and provides communication between each of the components in the electronic design system 10 during the design cycle, for example, to provide real time design rules checking. A functional simulator 38 can also be used to simulate and verify logic functionality of the circuit design represented by the architecture independent schematic representation 12.

The computer-implemented electronic design system 10 further includes a circuit component representation library 40 of pre-defined functional representations 14 of circuit components and corresponding architecture specific representations 20 existing as objects in the circuit component representation library 40. The functional representation schematic designer 30 can be used to build new functional representations 14 from the existing pre-defined functional representations 14 stored in the circuit component library 40. A target circuit architecture library 42 contains pre-defined target circuit architecture representations 22 that represent the target circuit architectures supported on the system 10.

In the preferred embodiment of the computer-implemented electronic design system 10, Fig. 2, the architecture advisor 36 is a centralized, multi-threaded database system that provides real time communication with each component in the computer-implemented electronic design system 10. One function of the architecture advisor 36 is to capture design parameters or attributes throughout the design process. In

one example, the architecture advisor 36 creates and displays user interfaces or forms, for informing the user of the design parameters, such as operational characteristics of the circuit, that need to be specified and for capturing these parameters. Examples of the user interfaces displayed by the architecture advisor 36 are shown in Fig. 3. In one example, these user interfaces can be called from other components of the system 10, such as the schematic designer 30, the functional simulator 38, or the dynamic architecture floorplanner 34.

The design information captured by the architecture advisor 36 is preferably checked immediately by the architecture advisor 36 to determine if inconsistencies have been entered or design attributes have been overspecified. This design information then becomes part of the architecture advisor centralized database 37 and is used in the design rules checking/error correction described below.

The architecture advisor 36 preferably includes a design rules checker (DRC) 44, for receiving user input via the schematic designer 30 and floorplanner 34 during the design process and for performing design rules checking in real time against the database of design information that has been captured. The DRC 44 checks the design as FLBs 14 are selected, arranged and interconnected and also when the architecture specific representations 20 are placed or moved on the target architecture representation 22, as will be described in greater detail below. In one embodiment, the architecture advisor 36 informs the user of the error and suggests a correction. In another embodiment, the architecture advisor 36 automatically corrects an error detected as a result of user input, for example, by reconfiguring the architecture specific representation 20, as will be described in greater detail below.

The architecture advisor 36 also preferably provides the user with access to the entire database to determine statistics, such as power consumption, routability, and static timing, for all or part of the design. Another function of the architecture advisor 36 is to interface with

the architecture specific configuration generator 32 to initiate the generation or re-configuration of architecture specific representations 20, as will be described in greater detail below.

According to the preferred embodiment, the schematic designer 30 is coupled to the architecture advisor 36 and provides design information to the architecture advisor 36 as the user selects, arranges and interconnects functional representations or FLBs 14 for each circuit component in the electronic circuit being designed. The architecture advisor 36 preferably notifies the schematic designer 30 in real time of any design errors as a result of improper selection, arrangement, or interconnection of FLBs 14 that do not satisfy the design criteria.

The preferred embodiment of the schematic designer 30 creates a schematic user interface that allows the user to select and arrange the FLBs 14 and interconnect the FLBs 14 with net segments 16 (see Fig. 1). One example of the schematic user interface displayed by the schematic designer 30 for creating the architecture independent schematic representation 12 is shown in Fig. 4. The schematic designer 30 can be used to select predefined FLBs 14 from the circuit component representation library 40 or can be used to create new FLBs from groups 18 of predefined FLBs 14. The circuit component representation library 40 preferably includes architecture independent functional data that defines the functions of each FLB 14. The schematic designer 30 also allows FLB parameters or attributes, such as types of inputs/outputs, the bitwidths, and the number of clock cycles (or latency), to be specified and displayed as annotations with the architecture independent schematic representation 12. When a predefined FLB 14 is selected using the schematic designer 30, the architecture independent functional data for that FLB is sent to the architecture advisor 36 together with the specified FLB parameters so that the architecture advisor 36 can initiate the translation of FLBs 14 to a target architecture, as described below. In addition to hardware FLBs representing

circuit components, the present invention contemplates software FLBs that represent software code running along side the hardware FLBs.

In one example, the schematic designer 30 creates the architecture independent schematic representation 12 of the electronic circuit as a hierarchy of FLBs 14. For example, new, higher level FLBs are created from groups 18 of existing FLBs 14. This allows users to build unique large scale functional blocks from predefined FLBs 14 that facilitate the design of complex electronic circuits. These large scale functional blocks can be parameterized and configured for selected target architectures in the same way as the individual FLBs 14 used to create them, as will be described in greater detail below. Thus, by re-targeting the individual FLBs 14 toward different or future architectures, the large scale functional blocks will be re-targeted without having to be redesigned from scratch for a different or new architecture. The large scale functional blocks are preferably generated using an API-based C++ process and use a standard Dynamically Loaded Library (DLL) interface, which allows the large scale functional blocks to be completely parameterizable. The DLL for a large scale functional block is called when the design is recompiled or when placement is moved on the target circuit architecture 22 and the large scale functional block is re-mapped. The large scale functional blocks can also be stored, for example, on a storage medium 39, for future use or distribution.

The preferred embodiment of schematic designer 30 also performs bit width alignment for signals (represented by net segments 16) connected to each circuit component represented by the FLBs 14. Thus, the user does not have to manage sign-extensions and ground-padding when two signals of different bit widths are connected to the same circuit component FLB. If the schematic designer 30 detects a bit-width mismatch that cannot be resolved, e.g. upon determination by the architecture advisor 36, the user is notified in real time of the error. The schematic designer

30 also performs bit width propagation, for example, when a bit-width mismatch is detected, bit widths can be automatically adjusted and propagated throughout the design.

According to the preferred embodiment, the functional simulator 38 is coupled to the functional representation schematic designer 30 and the architecture advisor 36, for simulating the functionality of the circuit represented by the architecture independent representation 12. The functional simulator 38 is preferably capable of handling multiple clocks and the interaction between data paths having multiple clocks. The functional simulator 38 also preferably supports half clock cycle operations. The functional simulator 38 can also provide dynamic pipeline realignment when one circuit component is replaced with another circuit component having a different latency. The functional simulator 38 can further provide dynamic power estimation based on a data stream.

The functional simulator 38 displays the signal status during simulation using a waveform viewer or using an X/Y graph or histogram. The functional simulator 38 also allows the user to build a fully interactive testbench by placing testbench components in the schematic representation 12 or by dynamically linking in a module which analyzes or further processes the simulation result using a DLL interface that passes data to a DLL function. In one example, custom test bench elements can be created, for example, using C++, and stored in a generic simulation black box library 45 for use by the functional simulator 38 to more efficiently analyze the simulation data.

According to the preferred embodiment, the architecture specific configuration generator 32 is a library management system coupled to the architecture advisor 36 for creating and managing the architecture specific representations 20 of the circuit components. The architecture specific representations 20 exist in the circuit component representation library 40 as configuration objects that define the configuration of each FLB on different target architectures. The configuration generator 32 can

dynamically load the configuration objects and preferably creates a cache 46, for storing the configuration objects corresponding to the FLBs 14 that are retrieved from the circuit component representation library 40. In one example, the cache 46 resides in RAM on a PC. The present invention also contemplates swapping the configuration objects to disk or re-generating the architecture specific representations 20 as needed. The use of cache 46 allows the architecture specific representations 20 to be more quickly loaded, configured and reconfigured, as will be described in greater detail below.

The configuration generator 32 is typically called by the architecture advisor 36 with a number of parameters, such as the FLB type (e.g., adder, multiplier, multiplexer), bit widths on inputs and outputs, speed requirements, architecture (site type), and the like. In response to the request from the architecture advisor 36 with the design parameters, the configuration generator 32 dynamically loads the appropriate configuration object using these parameters, for example, from the circuit component representation library 40 or from the cache 46. In one example, as the architecture independent schematic design 12 is generated, each of the FLBs are translated into a default target architecture by the architecture advisor 36. The configuration object corresponding to each FLB 14 for that default target architecture initially becomes the active configuration object that is available to the dynamic architecture floorplanner 34 for displaying as an architecture specific representation 20. As the FLBs 14 are moved on the target architecture representation 22, the active configuration object can change if necessary, as described below.

The configuration objects preferably include architecture specific configuration data 48 as well as configuration software routines 50 for configuring the configuration data 48 when called by the configuration generator 32. The architecture specific configuration data 48 defines the architecture specific representations 20 by

pre-mapping each FLB 14 to the architecture sites 24 of each of the target circuit architecture representations 22. Thus, according to this embodiment, the architecture specific configuration generator 32 acts as a collection of dynamically linked objects with a common interface. The architecture advisor 36 can use this interface to extract statistics from the configuration generator 32 pertaining to the available architecture specific representations 20.

According to the preferred embodiment, the dynamic architecture floorplanner 34 is coupled to the architecture advisor 36 to perform design rules checking and to automatically configure the architecture specific representations 20 when creating the architecture specific schematic representation. The dynamic architecture floorplanner 34 creates a user interface that allows the user to select and display one or more target architecture representations 22 including an array of the architecture sites 24 specific to the selected target architecture representation(s) 22. The target architecture library 42 preferably includes target architecture data defining each target architecture supported on the system 10. When one or more target architectures are selected by the user, the corresponding target architecture data is retrieved from the target architecture library 42 and the corresponding target architecture representation(s) 22 is displayed by the dynamic architecture floorplanner 34. Each selected target architecture also becomes an object in the architecture advisor 30 database 37, representing a collection of architecture sites that makes up that target architecture.

One example of the target architecture representations 22 displayed by the dynamic architecture floorplanner 34 is shown in Fig. 5. The architecture sites 24 represent the lowest logical granularity or breakdown of physical locations within a particular target architecture. With a Xilinx® series FPGA, a 10 by 10 array of Configurable Logic Blocks (CLBs) would be represented by a 10 by 10 array of architecture sites. Different types of architecture sites can also be used depending upon the architecture type, for

example, one type of architecture site 24a represents a CLB in the Xilinx® series FPGA while another type of architecture site 24b represents an Input Output Block (IOB) in the Xilinx® series FPGA. The architecture sites 24 are preferably assigned unique ID numbers representing their type and location on the target architecture representation 22.

The dynamic architecture floorplanner 34 displays architecture specific representations 20 on the selected target architecture representation 22 to create the architecture specific schematic representation. The architecture specific representations 20 represent the functionality of an FLB mapped to specific types of architecture sites 24, and each FLB 14 has a plurality of different architecture specific representations 20 for different types of architectures. As mentioned above, the architecture specific representations 20 represent configuration objects having configuration data that defines the type of architecture sites supported by that object. The dynamic architecture floorplanner 34 preferably interacts with the configuration generator 32 via the architecture advisor 36 to display the active architecture specific representation 20 corresponding to the active configuration object on the selected target architecture representation 22. The dynamic architecture floorplanner 34 partitions the active architecture specific representation 20 into configuration blocks and displays the configuration blocks as occupying at least a portion of the architecture sites 24 on the target architecture representation 22.

To place the FLBs 14 in the architecture independent schematic representation 12 on the target architecture representation 22, the system preferably creates a user interface in which the architecture independent schematic representation 12 and the target architecture representation 22 are displayed simultaneously, as shown in Fig. 6. This interface allows FLBs 14 or groups of FLBs 14 on the architecture independent schematic representation 12 to be moved to the target architecture representation 22 at a

selected location, for example, using a "drag and drop" technique.

When an FLB 14 or group 18 of FLBs 14 is moved to the selected location on the target architecture representation 22, the dynamic architecture floorplanner 34 determines the type of architecture sites 24 at that selected location, for example, based upon the unique ID associated with the site at that location. The architecture advisor 36 compares the type of architecture site 24 at the selected location with the type of architecture sites supported by the active configuration object corresponding to the FLB 14 being placed at the selected location. If the type of architecture sites 24 at the selected location match the type supported by the active configuration object, the dynamic architecture floorplanner 34 changes the shape of the FLB 14 (commonly referred to as morphing) to the active architecture specific representation 20, which is partitioned and placed accordingly.

If the type of architecture sites 24 at the selected location do not match the type supported by the active configuration object, the architecture advisor 36 reports the site type at the selected location to the configuration generator 32. The configuration generator 32 then searches for a matching configuration object for that FLB that will support the selected site type. If a matching configuration object is found, this matching configuration object becomes active, and the dynamic architecture floorplanner 34 changes the shape of the FLB 14 to the new corresponding active architecture specific representation 20. If no matching configuration objects associated with that FLB are found, either in the cache 46 or the library 42, the architecture advisor 36 and the user are notified that the FLB cannot be placed at this location.

The preferred embodiment of the dynamic architecture floorplanner 34 also allows the architecture specific representations 20 that have been placed on the target architecture representation 22 to be moved to new locations on the target architecture representation 22 or to a

different target architecture representation 22 for another type of architecture. When an architecture specific representation 20 is moved, the above process of finding a matching configuration object that supports the architecture sites at the new location is repeated. Thus, when an architecture specific representation 20 at one location is moved to a new location or to a new architecture, the active architecture specific representation 20 can change dynamically to a new architecture specific representation 20 appropriate for a new type of architecture site. Further, as architecture specific representations 20 are moved on the target architecture representation 22, the configuration objects corresponding to the architecture specific representations 20 can query the configuration objects corresponding to other architecture specific representations 20 on the target architecture representation 22.

Thus, the dynamic architecture floorplanner 34 can create a heterogeneous floorplanning environment across different architectures that supports other types of objects in addition to FLBs. This environment would allow design of heterogeneous processors, such as a FPGA and microprocessor combination, a FPGA and ASIC combination, or a FPGA and DSP chip combination.

In addition to being automatically configured for the type of architecture sites at which they are placed, the architecture specific representations 20 on the target architecture representation 22 can be configured or re-configured in response to real time design rules checking. The DRC 44 of the architecture advisor 36 detects errors in real time as the architecture specific representations 20 are initially placed on the target architecture representation 22 and/or when moved to new locations or new target architecture representations 22. The architecture advisor 36 is continuously informed of placement and configuration information via the configuration generator 32 and performs real time design rules checking against the database of design criteria to detect errors, such as timing or routing errors. In one embodiment, the dynamic

architecture floorplanner 34 receives any errors/warnings from the architecture advisor 36 and notifies the user in real time. When the architecture advisor 36 detects that this error has been corrected by the user, the user will also be notified of the correction. In another embodiment, the architecture advisor 36 can automatically correct design errors by causing the active configuration object responsible for the error to re-instantiate or re-configure in a way that alleviates the error.

The re-configuration is preferably performed by the configuration routines 50 within the configuration objects. As mentioned above, the configuration routines are in the form of software code associated with the architecture specific configuration data stored in the circuit component representation library 40 and made available by the configuration generator 32. The architecture advisor 36 preferably tracks the configuration objects available from the architecture specific configuration generator 32. The architecture advisor 36 also interacts with the configuration generator 32 and the available configuration objects to perform the design rules checking/error correction as the architecture specific representations 20 are placed and partitioned on the target architecture representation 22. This is preferably performed as a distributed, object oriented process that may involve dynamically loading configuration routines via the configuration generator 32. Thus, the user is always informed of what design choices are available at any given time and is immediately warned when design specifications are not being met. When a configuration routine is called, the routine uses information provided by the architecture advisor 36 to re-configure the architecture specific configuration data 48 within the configuration object.

In one example, an active configuration object is re-instantiated or re-configured based upon the proximity of the corresponding architecture specific representation 20 to other architecture specific representations 20 in the target architecture representation 22. The dynamic architecture

floorplanner 34 preferably displays the changing shape of the architecture specific representation 20 as it is moved across the target architecture representation 22. The dynamic architecture floorplanner 34 can also notify the user if the architecture specific 20 cannot be changed to that shape or location. One example of this type of dynamic re-configuration applied to an architecture specific representation 20 of a register being moved on a target architecture representation 22 is shown in Figs. 7A-7C.

In another example, an active configuration object is re-instantiated or re-configured to meet timing constraints, if possible. If the user creates too much routing distance between two architecture specific representations 20, for example, the design rules checker 44 will detect a routing error in real time. In warning mode, the dynamic architecture floorplanner 34 will display this error to the user. In auto correct mode, the new setup time requirements are passed to the configuration generator 32 via the architecture advisor 36 and the active configuration object will re-instantiate or re-configure itself to meet the required setup time. The dynamic architecture floorplanner 34 will dynamically reshape the corresponding active architecture specific representation 20 shown on the target architecture representation 22 accordingly. One example of the notification of set up time or routing errors is shown in Figs. 8A and 8B. One example of the dynamic re-configuration of an architecture specific representation 20 to correct set up time or routing errors is shown in Figs. 9A-9C. One example of the dynamic re-configuration of architecture specific representations within a group of FLBs is shown in Figs. 10A-10C.

As architecture specific representations 20 are moved on the target architecture representation 22, the architecture advisor 36 can also continuously monitor placement changes and recalculate static path timing and estimate routing in real time based upon the changes. This allows an incremental, real-time approach to static timing verification, providing instant feedback of incremental

changes to the floorplan of the architecture independent schematic representation.

The dynamic architecture floorplanner 34 also allows designs to be imported from other EDA design tools. The dynamic architecture floorplanner 34 can re-generate a hierarchy from a "flattened" design, facilitating the planning of that design on different target architecture representations 22. For example, conventional synthesized designs are "flattened" by the mapping stage in that the formats used to represent the mapped design do not support design hierarchy. The hierarchy, however, is preserved in the names of the various architecture components inside this port-map file format. The dynamic architecture floorplanner 34 of the present invention rebuilds the original design hierarchy based on the naming of the architecture components as well as the interconnect schemes. The dynamic architecture floorplanner 34 thus allows larger blocks of an imported design to be moved within the target architecture representation 22.

The dynamic architecture floorplanner 34 also allows entry of board topologies, for use as templates in mapping the algorithm to an existing piece of hardware. Board topologies are preferably stored in a reconfigurable board architecture library 52 and can be re-used when a different application is mapped to the same board.

The computer-implemented electronic design system 10 can be used with vendor router tools 60 and vendor post-route timing analysis tools 62. The computer-implemented electronic design system 10 further includes a requirements resource monitor 64, responsive to the vendor post-route timing analysis tools 62, for allowing the user to visualize and manage circuit timing information extracted from the post-route timing analysis tool 62. The computer-implemented electronic design system 10 can also be used with a board level schematic 66 and board-level simulator 68. The functional representation schematic designer 30 can access the board level schematic 66 while creating the architecture independent schematic representation of the

electronic circuit.

The present invention contemplates providing one or more of the components of the computer-implemented electronic design system apart from the other components of the system. The present invention contemplates different implementations of the functions performed by each of the components described in the exemplary embodiment above.

Accordingly, the computer-implemented electronic design system of the present invention provides an integrated system for designing an architecture independent schematic representation of an electronic circuit, such as a logic circuit, and for implementing that schematic representation on one or more target architectures. The computer-implemented electronic design system of the present invention also provides real time design checking throughout the design process and dynamic re-configuration of the design implementation on a selected target architecture to correct design errors and/or optimize the design.

Modifications and substitutions by one of ordinary skill in the art are considered to be within the scope of the present invention which is not to be limited except by the claims which follow.

What is claimed is:

CLAIMS

1. A computer-implemented electronic design system for designing an electronic circuit having at least one circuit component within a target circuit architecture, said system comprising:

an architecture advisor, for receiving design criteria and for creating a database of design criteria, and for monitoring and managing design of said electronic circuit using said design criteria;

a functional representation schematic designer, responsive to said architecture advisor, for creating and displaying an architecture independent schematic representation of said electronic circuit, wherein said architecture independent schematic representation includes a functional representation of each said at least one circuit component in said electronic circuit;

an architecture specific representation generator, responsive to said architecture advisor, for generating at least one architecture specific representation corresponding to said functional representation of each said at least one electronic circuit component when implemented on at least one target circuit architecture; and

a dynamic architecture floorplanner, responsive to said architecture advisor, for creating and displaying at least one architecture specific schematic representation of said electronic circuit, said architecture specific schematic representation including at least one target architecture representation having an array of architecture sites with said at least one architecture specific representation of each said at least one circuit component implemented thereon, wherein said architecture specific configuration generator generates said at least one architecture specific representation configured for said architecture sites of said at least one target architecture representation when said functional representation of said

at least one circuit component is placed at a location on said at least one target architecture representation.

2. The computer-implemented electronic design system of claim 1 wherein said architecture advisor includes a design rules checker, for performing real time design rules checking against said database of design criteria as said architecture independent schematic representation is created using said functional representation schematic designer and as said architecture specific schematic representation is created using said dynamic architecture floorplanner, and wherein said architecture advisor informs said user if a design error is detected.

3. The computer-implemented electronic design system of claim 2 wherein said architecture advisor automatically corrects said design of said electronic circuit if a design error is detected.

4. The computer-implemented electronic design system of claim 1 further including a circuit component representation library including a plurality of predefined functional representations of circuit components, and including a plurality of architecture specific representations associated with each of said plurality of functional representations.

5. The computer-implemented electronic design system of claim 4 wherein said circuit component representation library includes:

architecture independent functional data defining functions of said predefined functional representations; and

architecture specific representation data associated with said architecture independent functional data for each of said predefined functional representations, wherein said architecture specific circuit component configuration data defines an

implementation of said functions of said predefined functional representations on a plurality of target circuit architectures.

6. The computer-implemented electronic design system of claim 4 wherein said functional representation schematic designer allows said user to select and arrange selected ones of said plurality of predefined functional representations and to provide interconnections between each of said selected ones of said plurality of predefined functional representations.

7. The computer-implemented electronic design system of claim 6 wherein said functional representation schematic designer manages bit alignment for said interconnections provided between each of said selected ones of said plurality of predefined functional representations.

8. The computer-implemented electronic design system of claim 1 wherein said functional representation schematic designer displays circuit component attributes with said functional representation of each said at least one circuit component.

9. The computer-implemented electronic design system of claim 1 further including a target circuit architecture representation library including a plurality of target circuit architecture representations, wherein said dynamic architecture floorplanner allows said user to select at least one of said plurality of target architecture representations.

10. The computer-implemented electronic design system of claim 1 wherein said dynamic architecture floorplanner, in response to placing said functional representation of each said at least one circuit component at said location on said at least one target circuit architecture representation, notifies said architecture advisor of said

location, wherein said architecture advisor causes said architecture specific configuration generator to generate said at least one architecture specific representation appropriate for said location on said target circuit architecture representation.

11. The computer-implemented electronic design system of claim 10 wherein said dynamic architecture floorplanner allows a user to move said at least one architecture specific representation to a new location on said target circuit architecture representation, wherein said architecture advisor, in response to movement of said at least one architecture specific representation to said new location, causes said architecture specific representation to dynamically re-configure based upon said new location.

12. The computer-implemented electronic design system of claim 11 wherein said dynamic architecture floorplanner displays changes to said architecture specific representation as said user moves said architecture specific representation to said new location.

13. The computer-implemented electronic design system of claim 10 wherein said architecture advisor includes a design rules checker, for performing real time design rules checking against said database of design criteria based upon said location of said architecture specific representation on said target architecture representation, and wherein, if a design error is detected, said architecture advisor causes said architecture specific representation to re-configure to correct said design error.

14. The computer-implemented electronic design system of claim 1 wherein said dynamic architecture planner displays at least first and second target circuit architecture representations, and wherein said dynamic architecture planner allows a user to move said at least

one architecture specific representation from said first target circuit architecture representation to said second target circuit architecture representation, wherein said architecture advisor, in response to movement of said at least one architecture specific representation, causes said architecture specific representation to dynamically re-configure based upon said second target circuit architecture.

15. The computer-implemented electronic design system of claim 1 further including a cache, responsive to said architecture specific configuration generator, for storing a plurality of architecture specific representations.

16. The computer-implemented electronic design system of claim 1 further including a functional simulator, responsive to said architecture advisor, for simulating functionality of said architecture independent schematic representation of said electronic circuit.

17. The computer-implemented electronic design system of claim 16 wherein said functional simulator simulates performance at multiple clock speeds.

18. The computer-implemented electronic design system of claim 16 wherein said functional simulator simulates performance at half clock cycles.

19. The computer-implemented electronic design system of claim 16 wherein said functional simulator enables dynamic pipeline re-alignment of said at least one architecture independent schematic representation of said at least one electronic circuit component.

20. The computer-implemented electronic design system of claim 1 wherein said at least one electronic circuit component is a logic circuit component.

21. The computer-implemented electronic design system of claim 1 wherein said electronic circuit is a field programmable gate array including a plurality of logic circuit components.

22. A computer-implemented system for designing an architecture independent schematic representation of a logic circuit including a plurality of logic circuit components on a target circuit architecture, said system comprising:

an architecture advisor, for receiving design criteria pertaining to said logic circuit, for monitoring design of said architecture independent schematic representation of said logic circuit, and for informing a user of design information;

a circuit component representation library including a plurality of predefined functional representations of said plurality of logic circuit components, wherein said predefined functional representations are independent of said target circuit architecture; and

a functional representation schematic designer, responsive to said architecture advisor, for allowing said user to select, arrange and interconnect selected ones of said plurality of predefined functional representations of logic circuit components to form said architecture independent schematic representation of said logic circuit, and for displaying said architecture independent schematic representation of said logic circuit, wherein said architecture advisor performs design rules checking in real time as said selected ones of said plurality of predefined functional representations of logic circuit components are selected, arranged and interconnected to form said architecture independent schematic representation of said logic circuit.

23. The system of claim 22 further including a functional simulator, responsive to said architecture advisor and said functional representation schematic

designer, for simulating logic functionality of said architecture independent schematic representation.

24. The computer-implemented system of claim 22 wherein at least one of said plurality of predefined functional representations of logic circuit components includes a large scale functional representation of a combination of logic circuit components.

25. A computer-readable medium for providing at least one electronic circuit component representation for use in a computer-implemented electronic design system for designing an electronic circuit having at least one electronic circuit component within a target circuit architecture, said computer-readable medium comprising:

architecture independent functional data defining at least one function to be performed by at least one electronic circuit component independent of said target circuit architecture;

architecture specific configuration data associated with said architecture independent functional data for each said at least one electronic circuit component, wherein said architecture specific configuration data defines a plurality of implementations of said at least one function on a plurality of target circuit architectures; and

a configuration routine, responsive to a request from said computer-implemented electronic design system indicating a selected location for said at least one electronic circuit component on said target architecture, for configuring said architecture specific configuration data to create an architecture specific representation appropriate for said selected location of said at least one electronic circuit component on said target architecture.

26. The computer-readable medium of claim 25 further including:

a re-configuration routine, responsive to a user changing said selected location of said at least one

circuit component on said target circuit architecture, for re-configuring said architecture specific representation using said architecture specific configuration data based upon said changed selected location of said at least one electronic circuit component on said target architecture.

27. The computer-readable medium of claim 25 wherein said at least one circuit component representation includes a large scale representation of a combination of circuit components, wherein said architecture independent functional data defines a plurality of functions to be performed by each circuit component in said combination of circuit components, wherein said architecture specific configuration data defines a plurality of implementations of each of said plurality of functions on a plurality of target architectures, and wherein said configuration routine configures said architecture specific configuration data to create an architecture specific large scale representation appropriate for said selected location of said at least one electronic circuit component on said target architecture.

28. A computer-implemented system for dynamic placement and partitioning of an electronic circuit design on a target circuit architecture, said system comprising:

an architecture advisor, for monitoring and managing design of said electronic circuit based upon design constraints and design criteria;

an architecture specific representation generator, responsive to said architecture advisor, for generating at least one architecture specific representation corresponding to each said at least one electronic circuit component when implemented on at least one target circuit architecture; and

a dynamic architecture floorplanner, responsive to said architecture advisor, for creating and displaying at least one target circuit architecture representation having an array of architecture sites with said at least one

architecture specific representation of each said at least one circuit component implemented thereon, wherein said at least one architecture specific representation of said at least one electronic circuit component is partitioned to occupy a portion of said architecture sites of said at least one target circuit architecture representation depending upon placement of said at least one circuit component at a location on said at least one target circuit architecture representation.

29. The system of claim 28 wherein said architecture specific configuration generator stores a plurality of architecture specific representations in cache memory.

30. The system of claim 28 further including further including a target circuit architecture representation library including a plurality of target circuit architecture representations, wherein said dynamic architecture planner allows a user to select one of said plurality of target circuit architectures.

31. The system of claim 28 wherein said dynamic architecture floorplanner imports said electronic circuit design from a storage medium.

32. A computer-implemented method for designing an electronic circuit, said electronic circuit including at least one electronic circuit component disposed within a target circuit architecture, said method comprising:

capturing design parameters;

selecting and displaying a functional representation of each said at least one electronic circuit component, wherein said functional representation represents an architecture independent function performed by said at least one electronic circuit component;

creating an architecture independent schematic representation of said electronic circuit having said at least one electronic circuit design parameter by arranging

and interconnecting said functional representation of each said at least one electronic circuit component;

selecting and displaying a target architecture representation for said electronic circuit, said target architecture representation including an array of architecture sites;

creating an architecture specific schematic representation from said functional representations of each said at least one electronic circuit component, wherein said architecture specific schematic representation includes an architecture specific representation of each said at least one circuit component implemented on said selected target architecture representation, and wherein said architecture specific representation is configured for said architecture sites on said selected target architecture; and

checking design rules in real time during designing said electronic circuit according to said design parameters.

33. The computer-implemented method of claim 32 wherein the step of selecting said functional representation of each said at least one electronic circuit component includes creating a hierarchy of functional representations including at least one group of functional representations.

34. The method of claim 33 wherein the step of creating a hierarchy of functional representations including at least one group of functional representations includes creating a large scale functional representation of a combination of circuit components, and further including the step of storing said large scale functional representation of said combination of circuit components.

35. The method of claim 32 wherein the step of selecting said functional representation of each said at least one electronic circuit component includes selecting a

large scale functional representation of a combination of circuit components.

36. The method of claim 32 wherein the step of checking design rules includes monitoring the selection, arrangement and interconnection of said functional representation, and checking for inconsistencies with said design parameters.

37. The method of claim 32 wherein the step of creating an architecture specific schematic representation from said functional representations includes moving each said function representation to a location on said selected target architecture representation.

38. The method of claim 37 wherein said architecture specific representation is dynamically configured based upon said architecture sites at said location on said target architecture representation.

39. The method of claim 37 wherein the step of checking design rules in real time includes checking said location of said architecture specific representation against said design parameters, and further including the step of dynamically re-configuring said architecture specific representation in response to detecting an error.

40. The method of claim 37 wherein the architecture specific representation is dynamically re-configured based upon a proximity of said architecture specific representation to another architecture specific representation on said target architecture representation.

41. The method of claim 37 wherein the architecture specific representation is dynamically re-configured based upon a set up time error detected when said architecture specific representation is placed at said location.

42. The method of claim 32 further including the steps of:

changing a location of an architecture specific representation on said target architecture representation;

monitoring changes of said location of said architecture specific representation on said target architecture representation; and

re-calculating estimated routing in real time based upon said changes of said location of said architecture specific representation on said target architecture representation.

43. The method of claim 32 further including simulating functionality of said architecture independent schematic representation.

44. The method of claim 32 wherein said electronic circuit is a logic circuit having a plurality of logic elements.

45. The method of claim 44 wherein said logic circuit is a field programmable gate array.

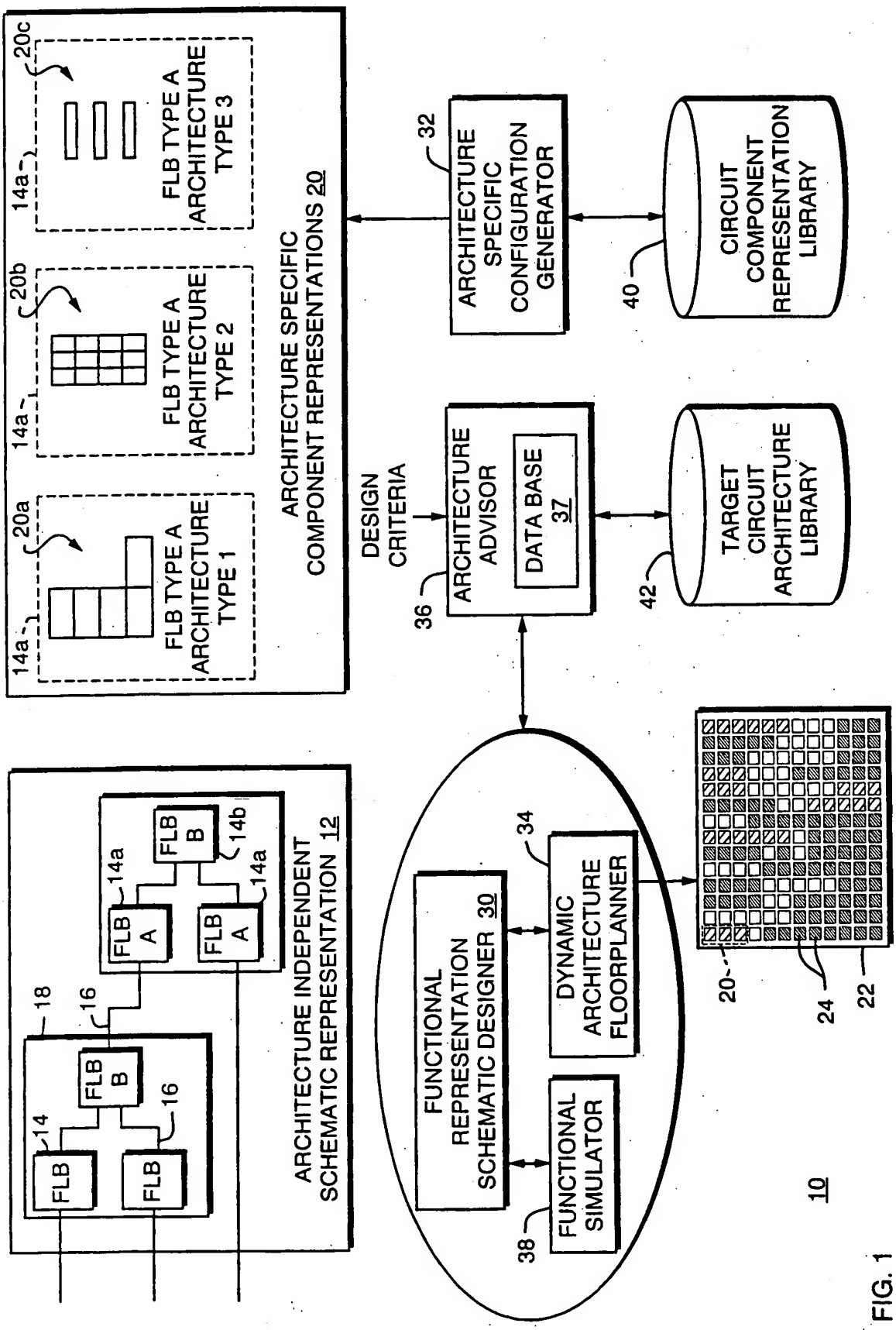


FIG. 1

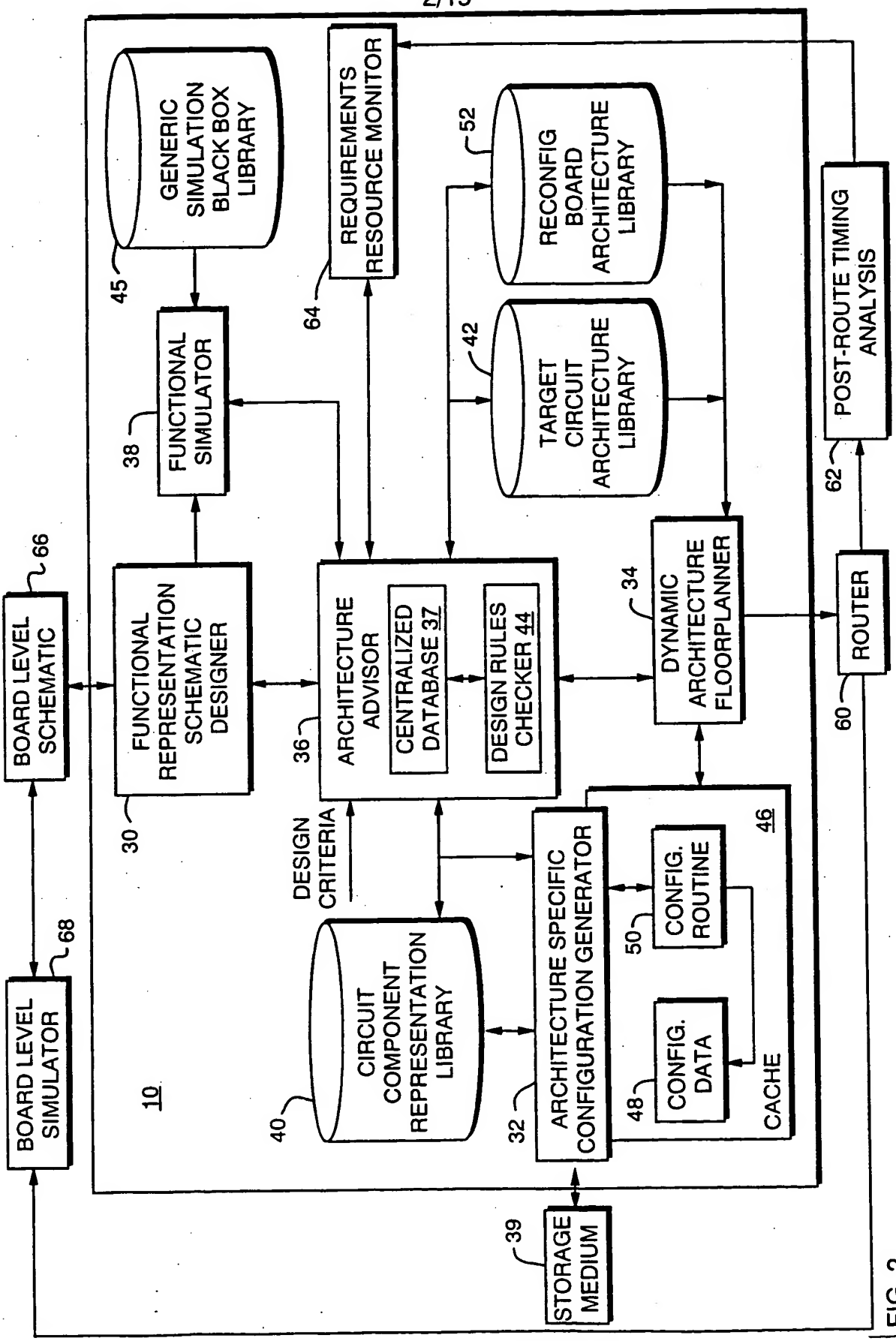


FIG. 2

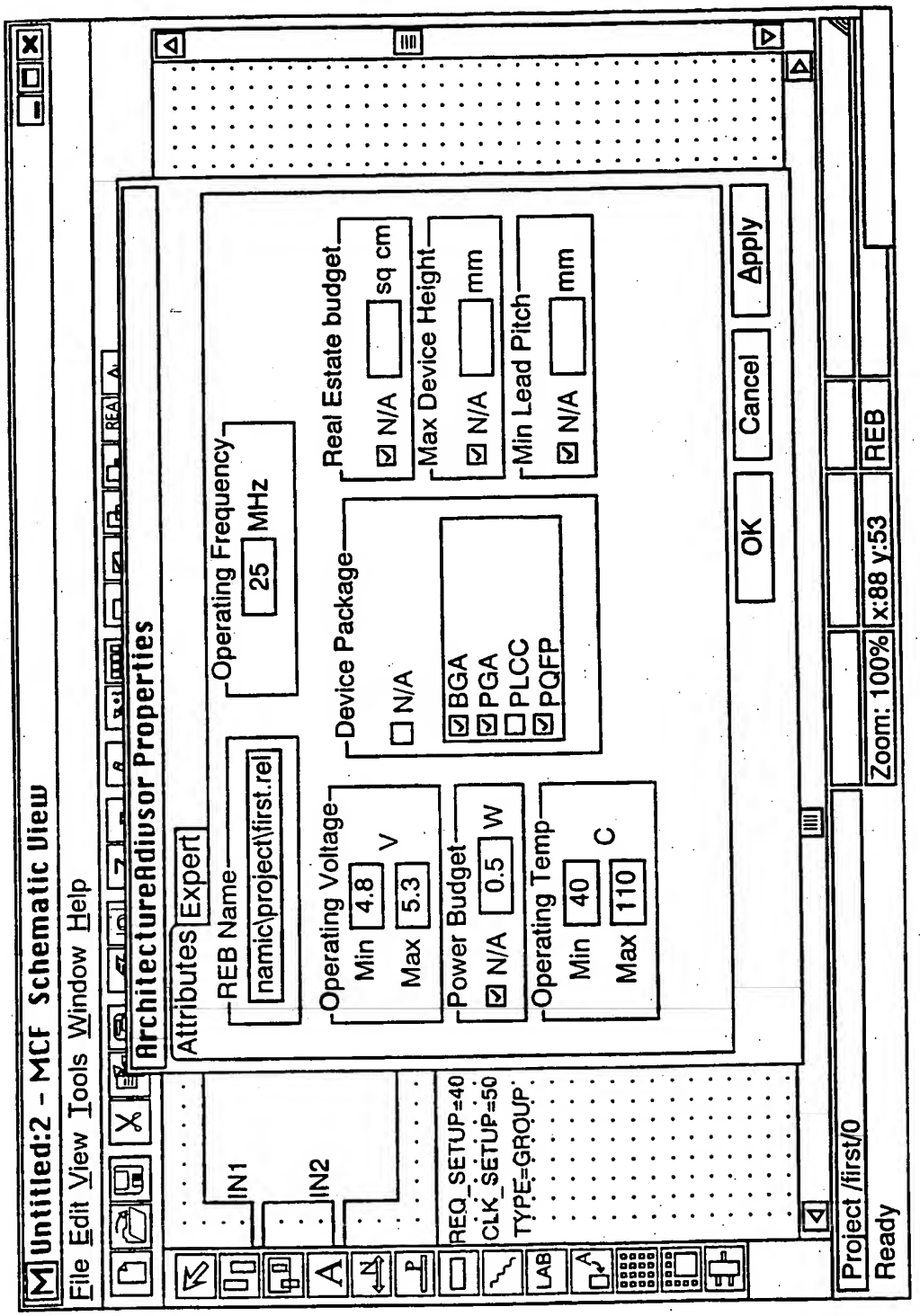


FIG. 3

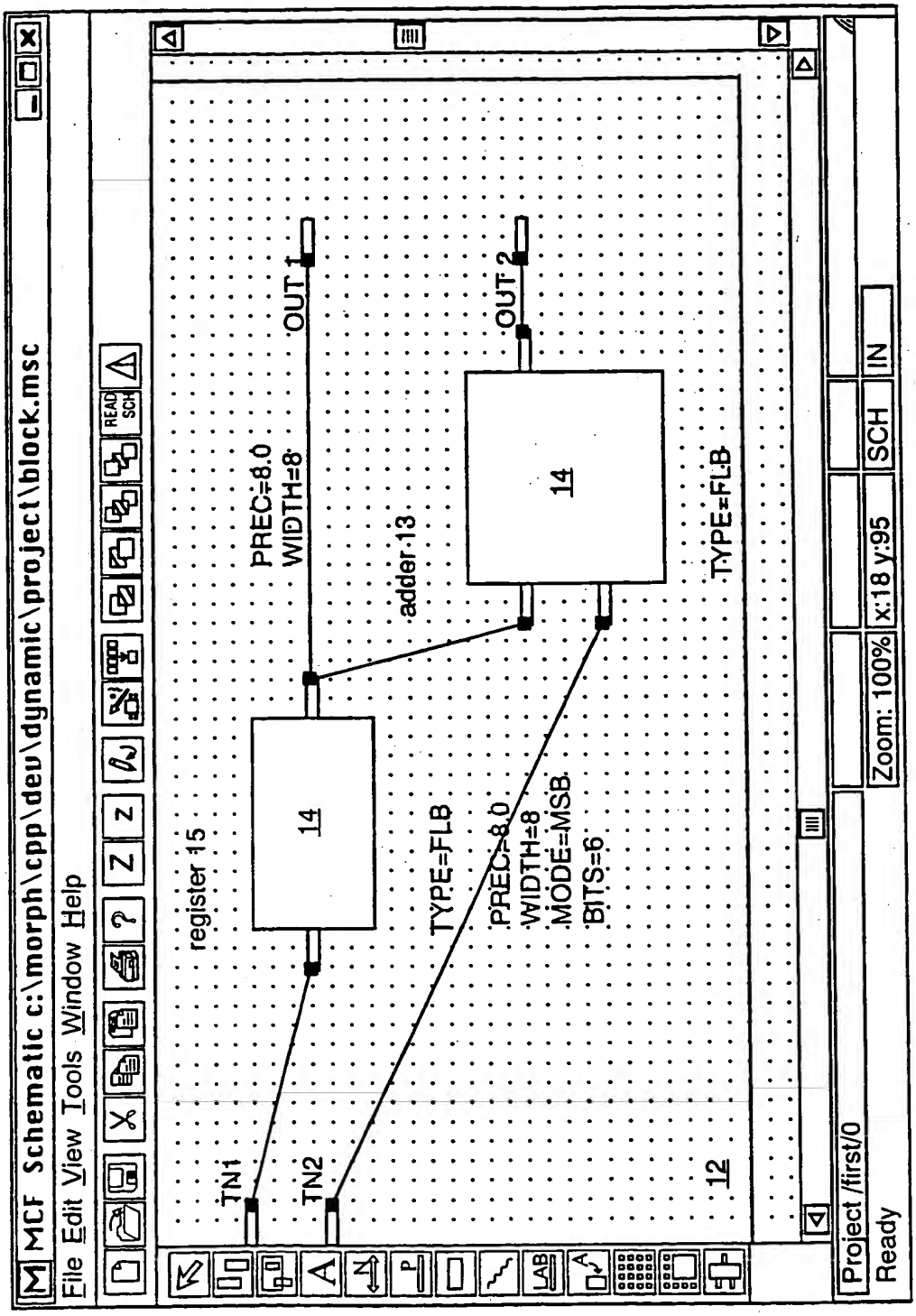
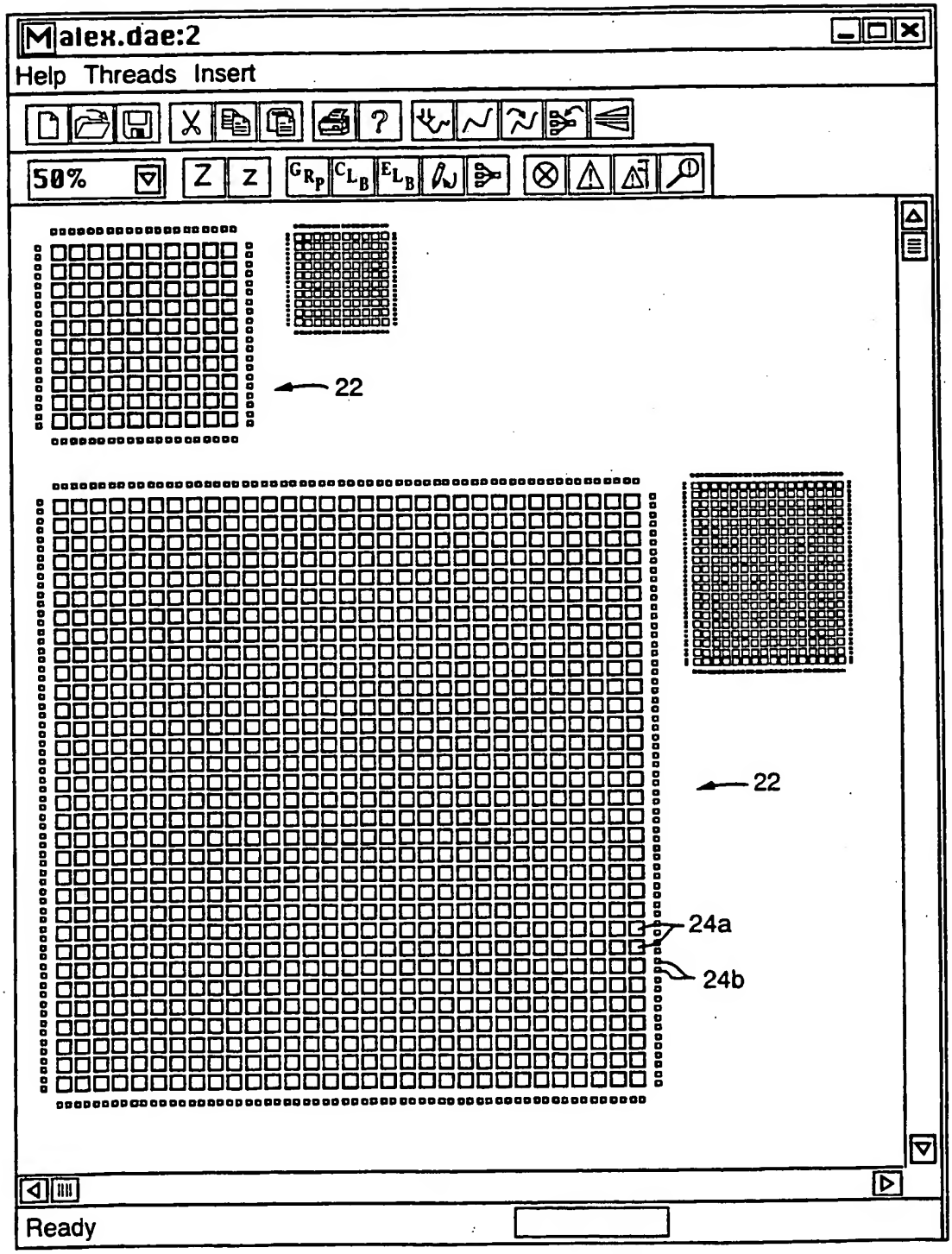


FIG. 4



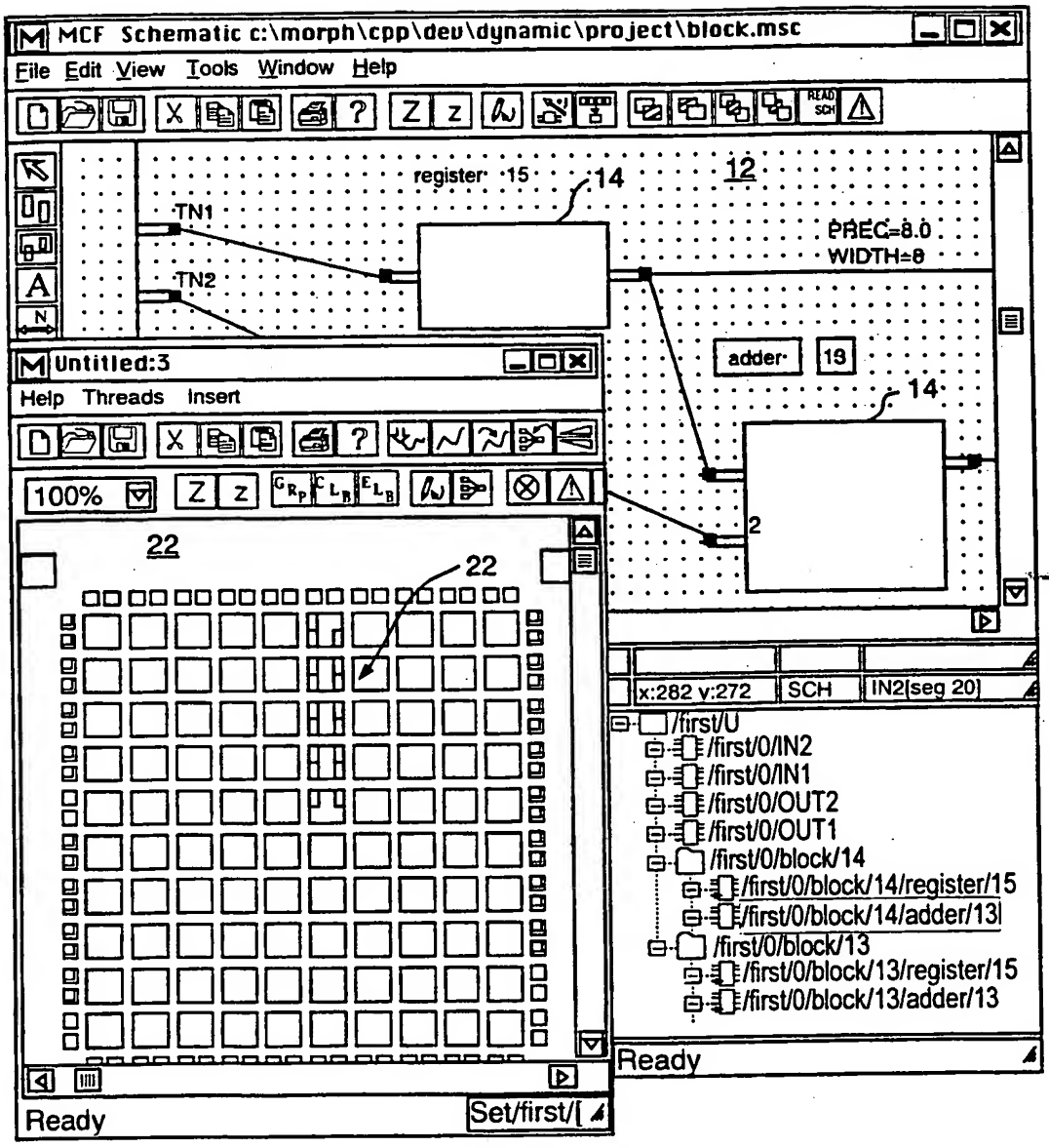


FIG. 6

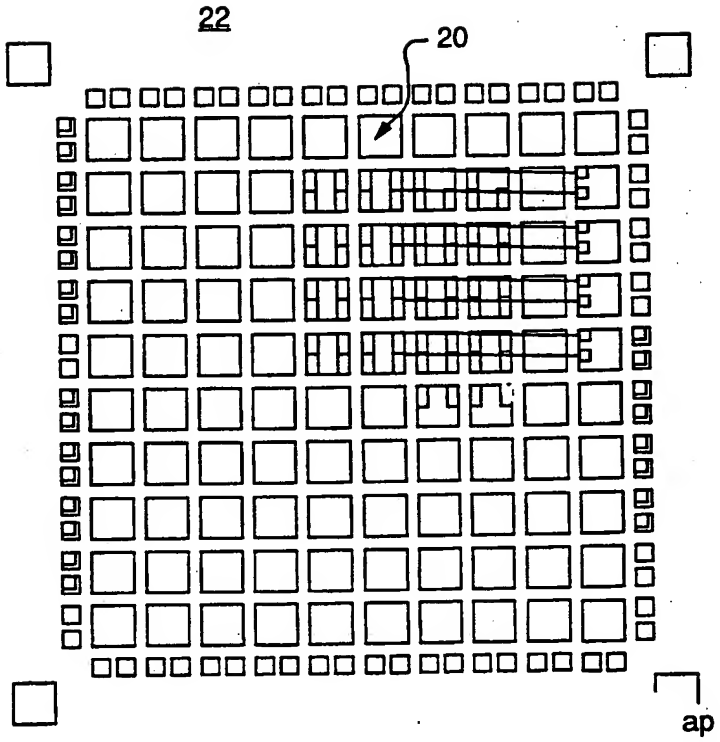


FIG. 7A

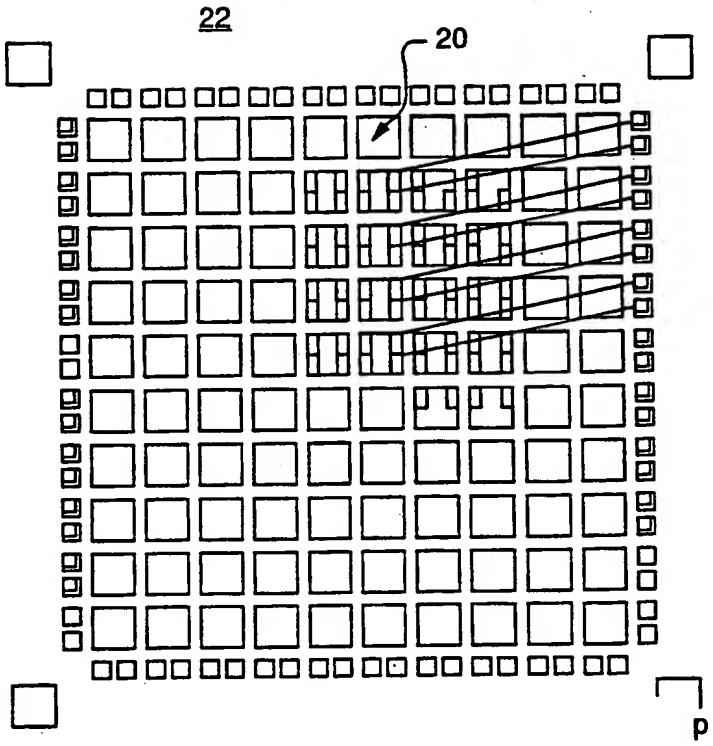


FIG. 7B

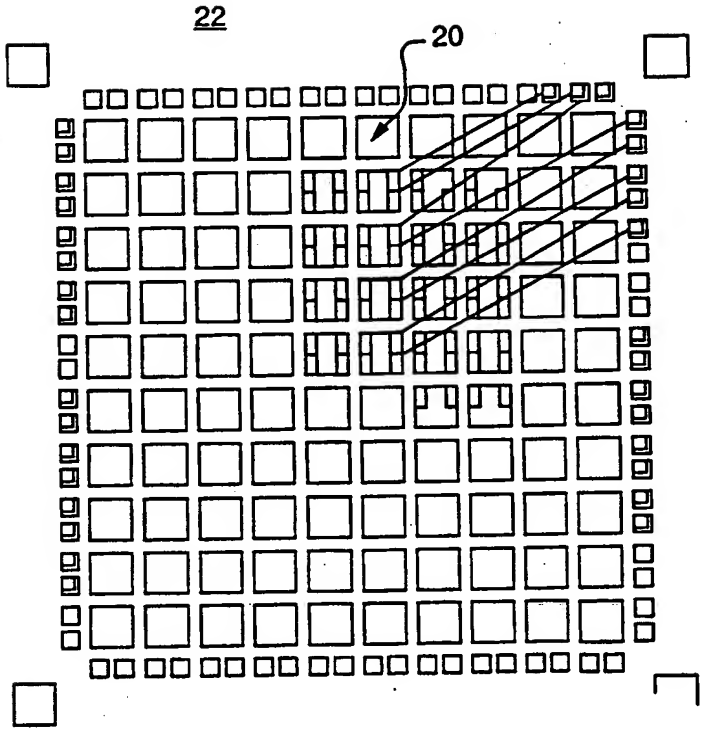


FIG. 7C

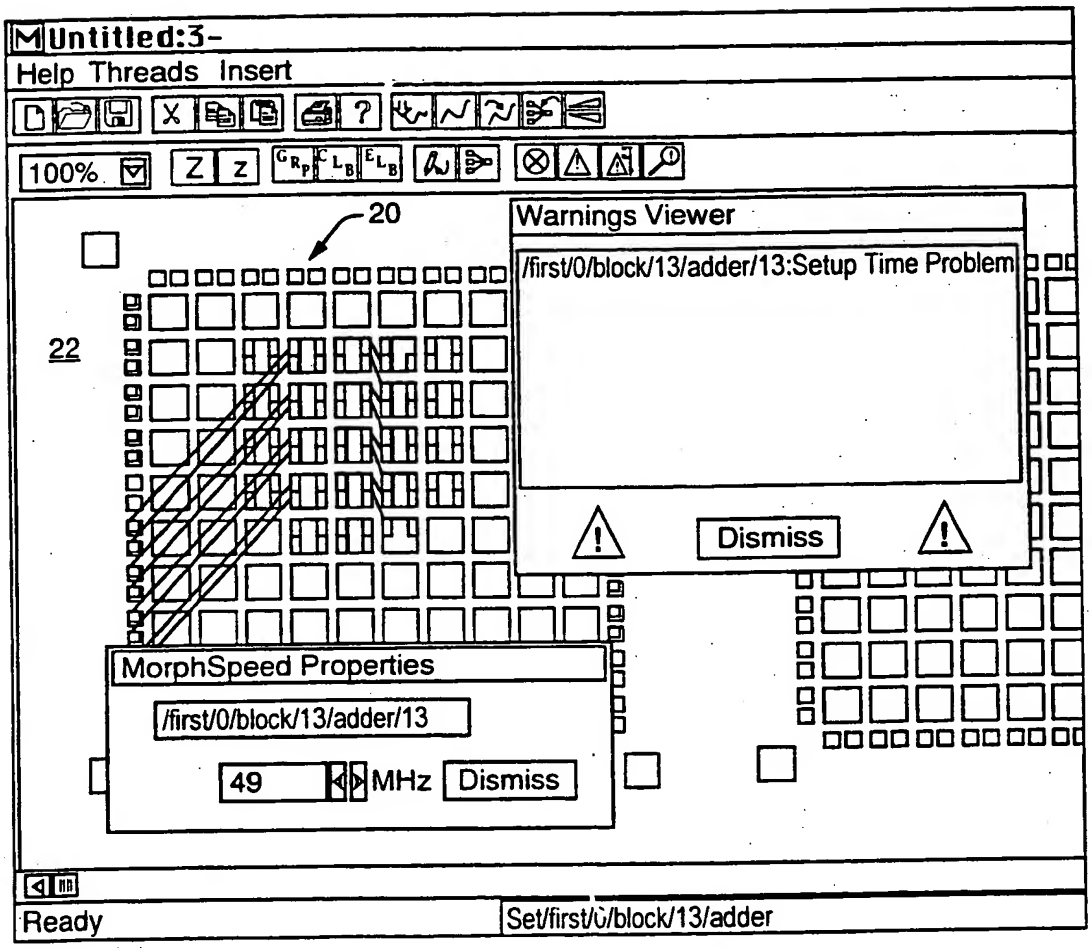


FIG. 8A

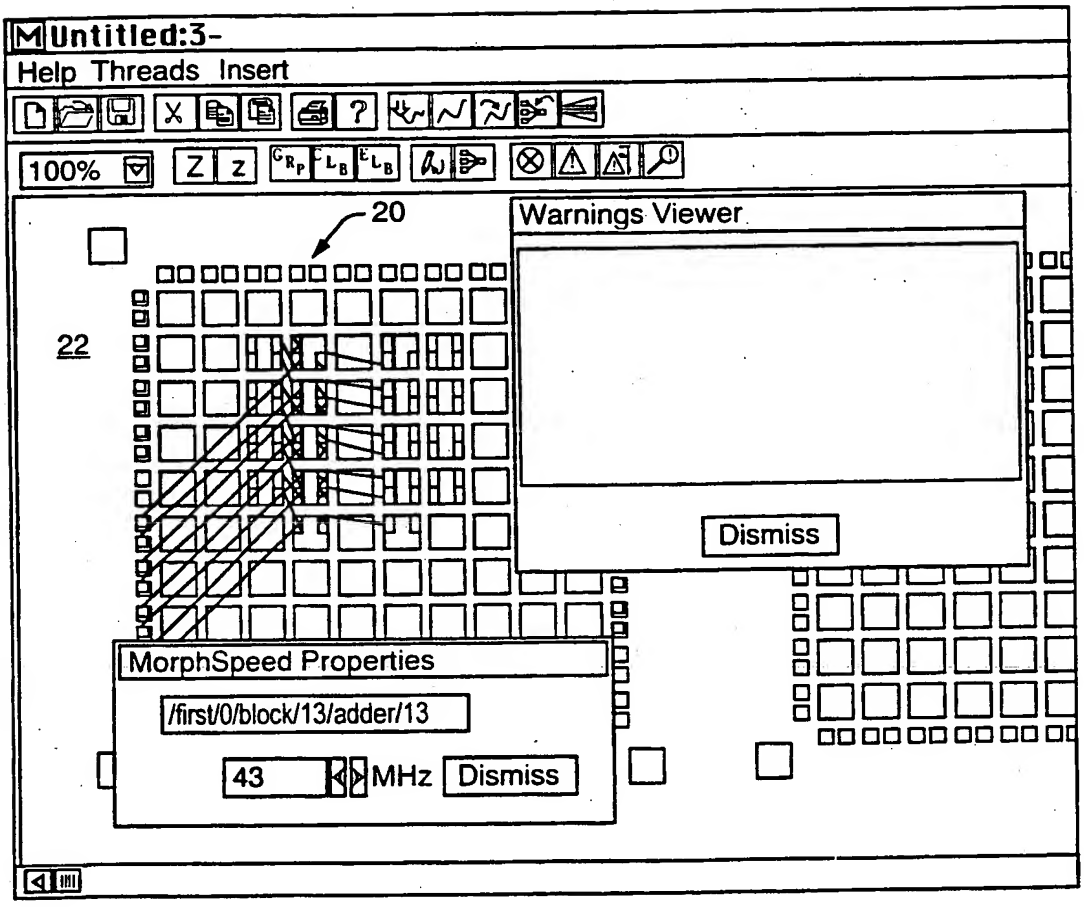


FIG. 8B

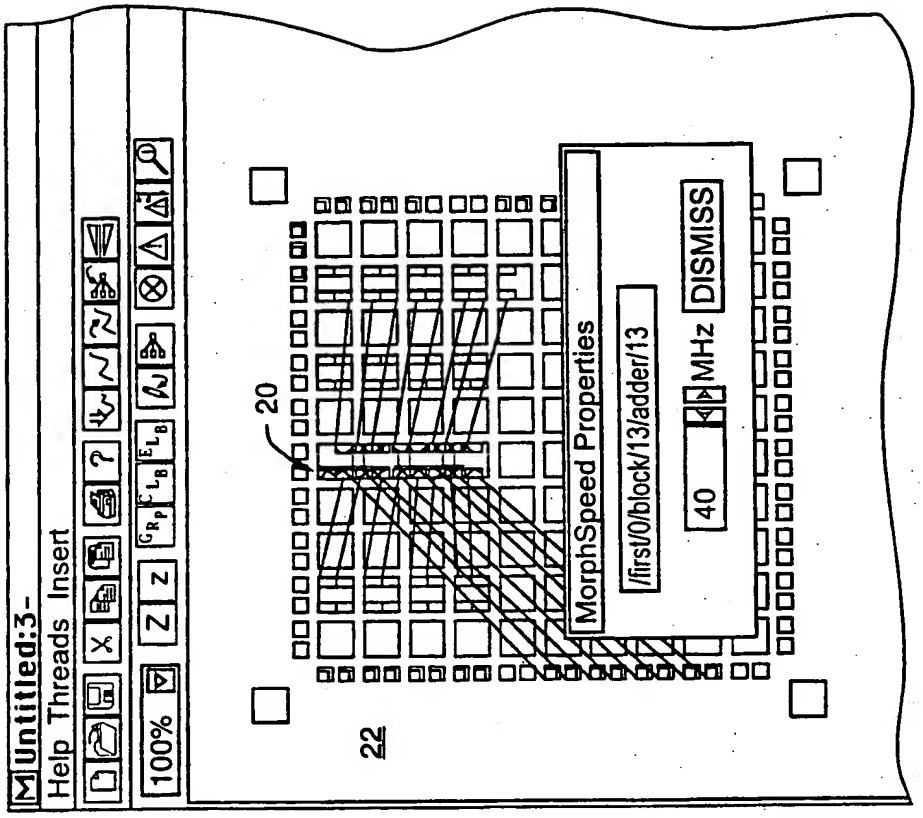


FIG. 9A

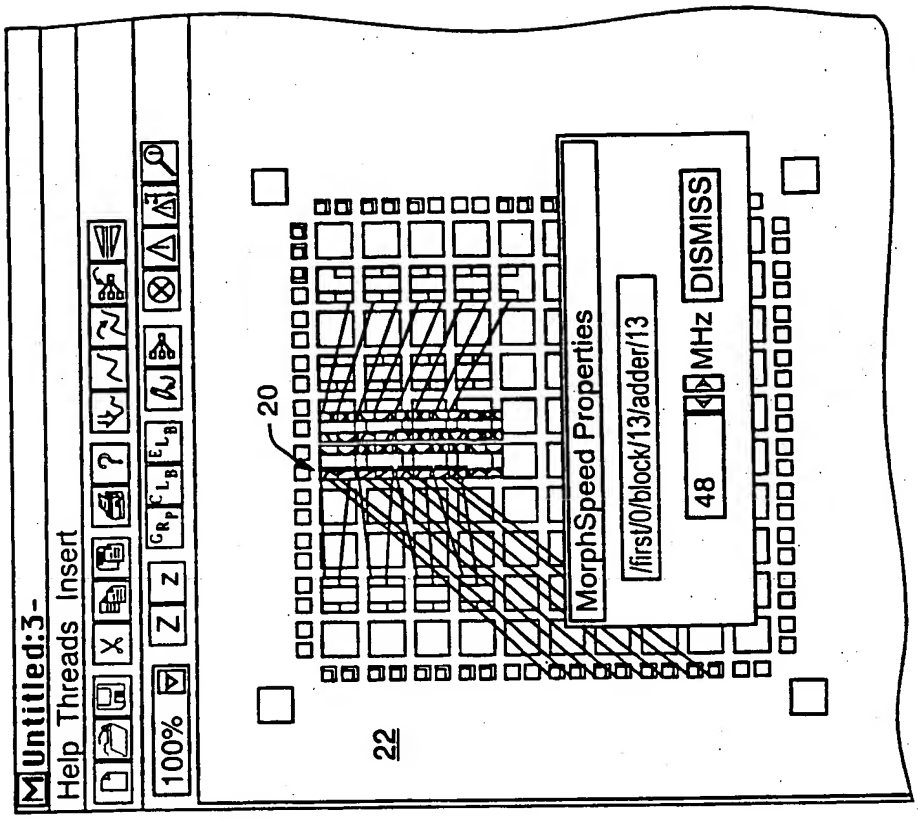


FIG. 9B

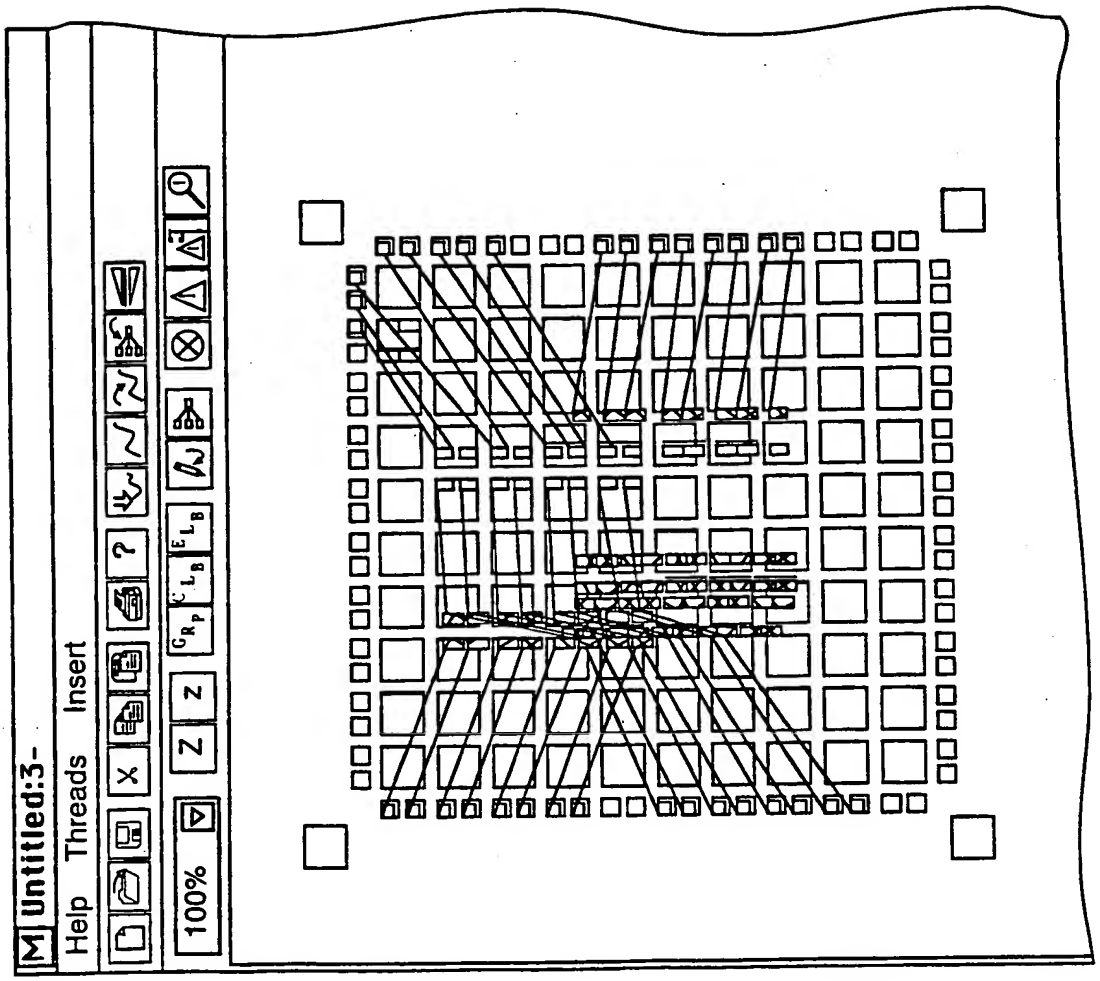
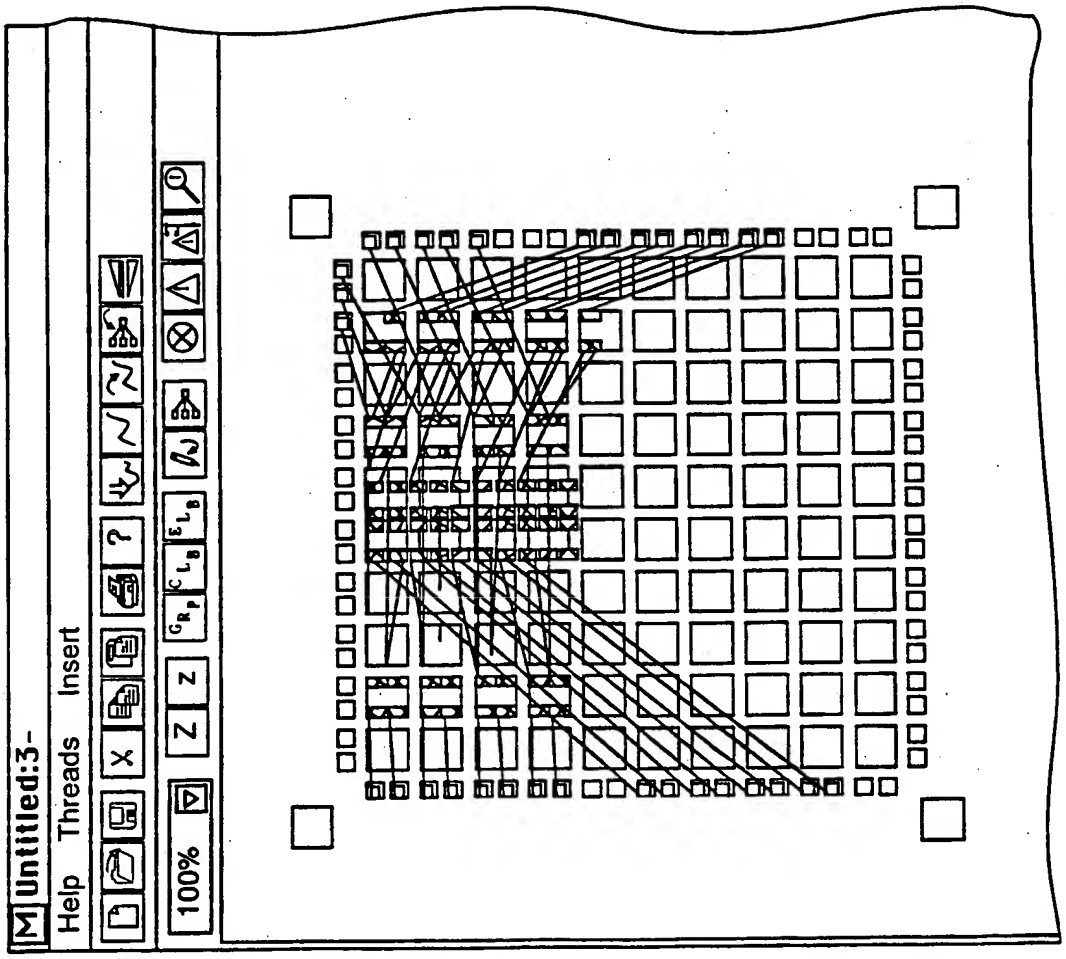


FIG. 10A



PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/50 — A3

(11) International Publication Number: WO 98/37475

(43) International Publication Date: 27 August 1998 (27.08.98)

(21) International Application Number: PCT/US98/02334

(22) International Filing Date: 6 February 1998 (06.02.98)

(30) Priority Data:
60/039,320 7 February 1997 (07.02.97) US

(71) Applicant: MORPHOLOGIC, INC. [US/US]; 131 Daniel Webster Highway #470, Nashua, NH 03060 (US).

(72) Inventors: MARTUSCELLO, Anthony, R.; 11 Oak Street, Dover, NH 03820 (US). BARBA, Alexandru; 131 Daniel Webster Highway #405, Nashua, NH 03060 (US). BOX, Brian; 131 Daniel Webster Highway #327, Nashua, NH 03060 (US). FURCINITI, Charles; 25 Proctor Road, Bedford, NH 03110 (US).

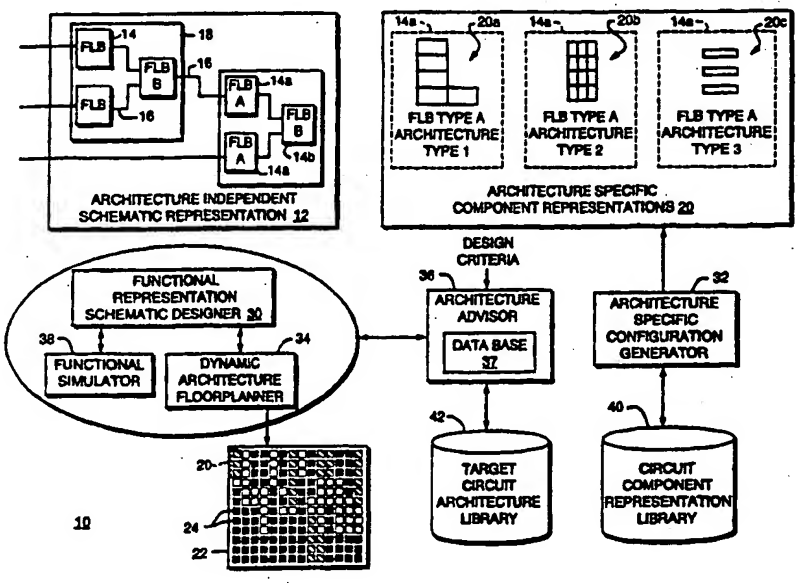
(74) Agents: BOURQUE, Daniel, J. et al.; Law Offices of Daniel J. Bourque, PA, Suite 303, 835 Hanover Street, Manchester, NH 03104 (US).

(81) Designated States: CA, JP, Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(88) Date of publication of the international search report:
18 February 1999 (18.02.99)

(54) Title: SYSTEM AND METHOD FOR DESIGNING ELECTRONIC CIRCUITS



(57) Abstract

A computer-implemented electronic design system (10) is used to design electronic circuits, such as field programmable gate arrays (FPGAs) and other complex logic circuits. The system allows a user to create an architecture independent schematic representation (12) of a circuit by selecting, arranging and interconnecting functional representations (14) of the circuit components. The system automatically configures the functional representations (14) for a selected target circuit architecture by generating architecture representations (20) of the circuit components for placement on the selected target architecture representation (22). The architecture specific representations (20) placed on the selected target architecture representation (22) can be moved to new locations or to another selected target architecture representation (22) and are dynamically re-configured for the new location or new target architecture.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/02334

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :G06F 17/50
US CL :364/491
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/488-491, 578; 326/41, 47; 395/919-921; 438/129

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, DRLINK

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 5,544,067 A (ROSTOKER et al) 06 AUGUST 1996, especially Figures 2-5, col. 6-11, 15-18, 19-20; col. 4, line 59 to col. 5, line 5; col. 29, lines 27-32; col. 26, lines 4-24; col. 14, lines 58-68, col. 9, lines 38-57; col. 12, line 61 to col. 13, line 16.	1-14, 16-28, 30-45 --- 15,29
Y	US 5,557,533 A (KOFORD et al) 17 SEPTEMBER 1996, col. 17-18.	15, 29

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
17 APRIL 1998	06 OCT 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer EMANUEL TODD VOELTZ Telephone No. (703) 305-9714

Form PCT/ISA/210 (second sheet)(July 1992)*